§2.2, 2.5, 2.11: The Anatomy of an Infant Program

devo

- *HW due at start of class*
- *Quiz ch1 back*

13 Sep 2006
CMPT14x
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Review from §1.8-2.1

- Expressions and precedence
  http://docs.python.org/ref/summary.html
- Five abstract components of hardware
- Software: instructions, languages, programs, OS
- Designer -> coder -> compiler -> assembler/linker
- Five control/structure abstractions of programs
- Pseudocode
- Importing library functions

A note about IDLE: subprocess

- Be sure to start IDLE from the Start menu, using File→ Open to edit your files
  - The right-click context menu "Edit with IDLE" starts IDLE in a slightly different mode
  - You should see "=== RESTART ===" each time you press F5 or do Run→"Run Module"
- Details for those interested:
  - "-n" command-line option causes IDLE not to use a separate subprocess for each run
  - Using the "-n" option will cause problems for you later when creating your own modules

Bugs and debugging

- Project stays “90% done” for 90% of the time
- Debugging takes up most of your time; allocate time for it!
- Spend a little more time on design and you'll save a lot of time debugging
- Syntax errors are easy to catch (compiler helps)
- Semantic (logical) errors come from poor design:
  - Much harder to catch, let alone fix!

What's on for today (§2.2, 2.5, 2.11)

- Components of a baby Python program
- Modules
- Library tools
- Literals, identifiers and reserved words
- Strings, quoting, newlines
- Statically-typed vs. dynamically-typed
- Declaring and initializing variables
- Keyboard input

Components of “helloworld.py”

```
"""A baby Python program.

Name: John Doe
This is a sample program.
"""

import math

print “Hello World!”
print “Pi =”, math.pi
```

Module docstring

Import library modules

Program statements

Modules

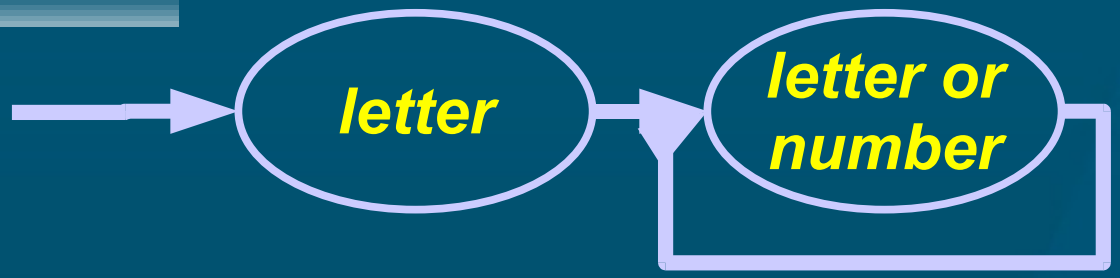
- A module is a container holding
  - items and information
    - Variables, functions, etc.
  - constituting all or part of an executable program
- helloworld.py is a module that is a complete executable program
- math is a library module from which we imported the pi constant
- math.pi is not a module but a name within a module

Identifiers

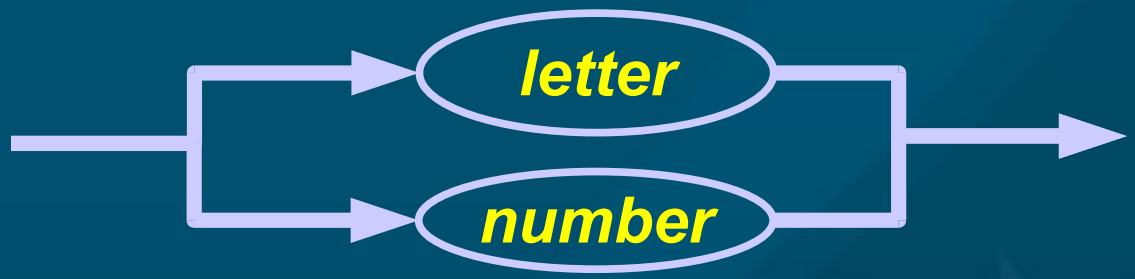
- Identifiers are names for stuff: e.g.,
  - Libraries (“math”)
  - Functions (“print”)
  - Variables (“numApples”)
- Identifiers are sequences of
  - non-blank letters or digits
  - Must start with a letter *(underscore _ counts as a letter)*
- OK: Great_Googly_Moogly, x, My21stBirthday
- Not OK: “hi ya”, h@Xz0r, 21stBirthday
- Case sensitive! Print ≠ print
- These are the rules; we'll talk about style tomorrow

Railroad diagram for identifiers

- identifier =



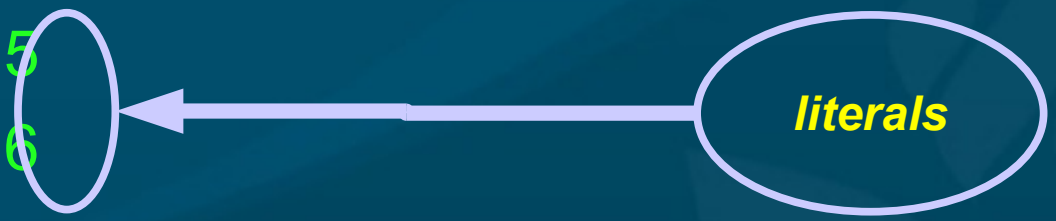
- letter or number =



- number = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- letter = {a, b, ..., z, A, B, ..., Z, _}

Literals vs. identifiers

- A literal is an entity whose name is an encoding of its value:
  - 187.3
  - “Hello World!”
  - True
- In contrast, the value of a variable may change even though its name stays the same:
  - numApples = 5
  - numApples = 6



Reserved words

- You can name your modules, functions, and variables almost anything you want, except
- Reserved words (keywords): special words or markers used to outline the structure of a program
  - import, if, else, while, for, def, ...
  - Complete list at http://docs.python.org/ref/keywords.html

Python Standard Library

- Library functions provided with every standard Python implementation
- You still have to import them, though
- Our HelloWorld.py program used pi from the math standard library

- There are oodles of standard library functions: http://docs.python.org/lib/lib.html

Strings and quoting

- Strings in Python can be in either 'single' or “double” quotes
- What if you want a quote mark in your string?
  - “It is I; don't be afraid”
  - 'Jesus said, “I am the way, and the truth, and the life.” '
- To include a newline (carriage return) in string, use three double-quotes:
  - ```
    """ This is a multi-line string.
    Even the newline is part of the string."""
    ```
  - This is rather special to Python; in M2 newlines just aren't allowed in strings
    ```

Splitting up strings: print

- print “Therefore go and”
  print “make disciples”
  - Therefore go and
    make disciples
- print “Therefore go and”,
  print “make disciples”
  - Therefore go and make disciples

Note trailing comma

Variables: names and values

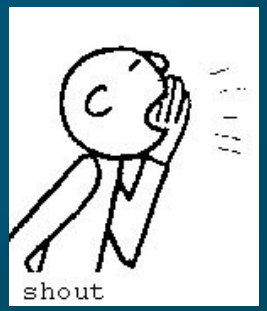
- A Python variable is a name for a memory location, the contents of which can be changed by a program.
  - numApples
- The assignment operator = is the means by which the name on the left is given the value on the right.
  - numApples = numApples + 1

Static typing vs. dynamic typing

- All variables have a type:
  - int, float, str, bool, etc.
- Some languages (e.g. C, M2, Java) are statically-typed:
  - Must declare the type of the variable ahead of time:
    - x, y : REAL;
    - k : CARDINAL;
  - Can't change type or assign a value of different type:
    - x := “Hello World!”; (* won't work in M2 *)
- But Python is dynamically-typed: variables can change type:
    - x = 5.0
    - x = True # okay in Python

Declaring vs. initializing

- This is only necessary for statically-typed languages:
  - Declare a variable to tell the compiler the type of the variable:
    - VAR numApples : CARDINAL; (* M2 *)
  - Its value is undefined until it is initialized:
    - BEGIN
      - numApples := 5; (* M2 *)
- In a dynamically-typed language like Python, just initialize the variable:
    - numApples = 5 # okay in Python



Keyboard input

- You know how to output using print()
- Use input() to get a value from the user:
  - balance = input("Opening balance? ")
  - The argument is the prompt string
  - Dynamic typing: Python interprets the user's response and determines its type
  - Just pressing Enter w/o input gives an error
- You can use raw_input() at the end of your program to wait for the user to press Enter before the program finishes

Review of today (§2.2, 2.5, 2.11)

- Components of a baby Python program
- Modules
- Library tools (what are some we know already?)
- Literals, identifiers and reserved words (examples?)
- Strings, quoting, newlines
- Statically-typed vs. dynamically-typed
- Declaring and initializing variables
  - (which are needed in C? In Python?)
- Keyboard input

TODO items

- Python/IDLE intro today (nothing to hand in)
- Homework due Friday:
  - §1.11 # 35
- Reading: through §2.4 for Thu; §2.8 for Fri
- Quiz ch2 next Mon
- Lab 1 due next MTW in lab section